Bi-manual haptic interaction in virtual worlds

Clément Nicolas

Bi-manual haptic interaction in virtual worlds

Clément NICOLAS - Master in Computer Science Research - ISTIC

Supervisors: **Anatole Lécuyer, Maud Marchal and Gabriel Cirio**

Team Bunraku/VR4I - IRISA

June 2011

Abstract

In the Virtual Reality field, force-feedback interfaces called haptic interfaces can simulate tactile and kinesthetic interactions. Bi-manual haptic interaction can better immerse users in virtual worlds than one hand interaction and more tasks can be realized such as parallel or precision tasks. Only a few studies deals specifically with bi-manual haptic interaction and previous work mainly extends uni-manual techniques directly to two hands. The document reports possible lacks of bi-manual-specific management of real and virtual workspace and the lack of genericity of solutions using haptic interfaces.

The study on bi-manual haptic interaction led to the realization of a framework allowing to use simultaneously several haptic devices. This framework simulates a 3D virtual world coupled with a physical simulation. We realized new specifically bi-manual haptic interaction techniques allowing to control camera, to extend the virtual workspace by a hybrid position/rate control and to help bi-manual pick and place task. The document point out issues such as collision between haptic devices and unification of two different haptic interfaces.

Contents

Chapter 1

Introduction

Every day we use our two hands to realize tasks: picking fragile or big objects, typing on a keyboard, holding a book while flipping pages, etc. Most of the time, we naturally use two hands for simplicity, safety or accuracy reasons, for the parallelization of two uni-manual tasks or bi-manual-only task. In the Virtual Reality field, force-feedback interfaces called haptic interfaces can simulate tactile and kinesthetic interaction, such as one-hand ore two-hand manipulations in virtual worlds.

The topic of my internship, "Bi-manual haptic interaction in virtual worlds", proposes to introduce the use of two haptic interfaces simultaneously. The purpose of the work is to study the unexplored possibilities of using two hands in virtual worlds with haptic feedback. This internship has led to a bi-manual haptic system shown in figure 1.1 connected to a virtual world and to new interaction techniques.

Figure 1.1: Our bi-manual haptic system

1.1 Presentation of the internship

Bunraku is a team of INRIA (Institut National de Recherche en Informatique et en Automatique) and IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires), in Rennes, working on virtual reality simulations and interaction with virtual worlds. The main challenge of the Bunraku team is to enable interactions between human and shared virtual environments. Bunraku investigates physically based models to represent the virtual environment, behavioral models to represent the virtual humans and multi-modal interaction models to express natural activity into virtual worlds.

This team has recently split into three different teams: MimeTIC (virtual human simulation), FRVSense (Rendering) and VR4I (interaction and collaboration in virtual worlds). My internship is realized in the VR4I team under the responsibility of Anatole Lécuyer, Maud Marchal and Gabriel Cirio.

The team realized several studies on haptic interaction in virtual worlds such as developing interaction metaphors, simulating haptic fluids, etc. Most of applications were uni-manual. A recent demonstration on haptic fluids ([CMHL10]) raised the question of using two hands in virtual reality. An analysis of existing works on that topic has shown that this field was recent and only a few number of articles was published. Bi-manual haptic devices have been designed and tested but some issues, such as collision between interfaces, have not been solved. Furthermore, no dedicated studies on bi-manual haptic interaction are present in the literature.

My first goal was to realize a software layer in order to work with multiple kinds of haptic devices in a virtual world. Then, using that framework, I designed new bi-manual haptic interaction techniques addressing some major issues. At the end of this internship we plan to validate the interaction techniques during a user-study experiment.

In this report we will first make a short glossary on the important terms used in the document. In Chapter 2, we will describe the existing work on bi-manual haptic interaction in virtual world and point the lacks that motivated the orientation of the internship. The chapter 3 will focus on the software and hardware conception describing the different choices, the structure of the solution and problems encountered. Chapter 4 will present the interaction techniques proposed and realized during the internship. In this part, a presentation of the challenges will be followed by the solutions proposed and the results obtained at this point of the internship. Finally we will make a conclusion about the realized work and present the perspectives planned for the rest of the internship.

1.2 Virtual reality

Virtual reality (VR) is a term proposed by Jaron Lanier in 1985. As defined in [PF06], VR is the field of computer generated simulation that can provide a sensory-motor activity in an artificial world. This world can be imaginary, symbolic or a simulation of the real world. Figure 1.2 shows a simple representation of Virtual Reality as an interaction between user(s) and a virtual world.

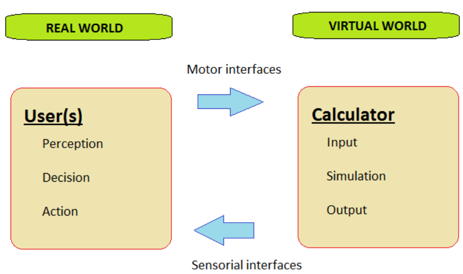

Figure 1.2: Perception, cognition, action loop in virtual world (from [PF06])

VR can change time representation, place or type of interaction. It can be achieved with informatics and behavioral interfaces in order to simulate in a virtual world the behavior of three dimensional entities. These entities are in real time interaction between them and user(s) in a pseudo-natural (simplified simulation of natural) immersion via sensory-motor channels [PF06].

1.3 Haptic interfaces

The human senses are not limited to the classic "five senses". Internal senses are less known. The senses related to the perception of our body, his movements, balance (through inner ear) or efforts (through muscles) are named proprioceptive senses. Haptic feedback uses both the tactile feedback, basically through the skin and kinesthetic sense. Kinesthetic information is sent via sensory systems by tension and compression of muscles in the body, tendons and joints. This sense allows manipulation and active exploration contrary to the five other senses [MO08].

A haptic interface is principally composed of a haptic device that can return to the user a haptic feedback. Figure 1.3 shows an example of a haptic device: the Virtuose 6D (Haption). This is a classical example of a six Degrees Of Freedom (DoF) haptic device. The probe can be positioned (translation on the three axis X,Y and Z; three DoF) and oriented (rotation around the three axis X,Y and Z; three DoF) allowing a haptic simulation of virtual tool in a Virtual Environment (VE).

Figure 1.3: A Virtuose 6D haptic device

As seen in Figure 1.4, the haptic device is connected to a calculator (in green) that process a haptic rendering. The haptic device sends information about its position, interpreted by the calculator that returns a haptic feedback to the user [PF06, MO08]. The user can thus perceive shapes, texture and stiffness of virtual objects.

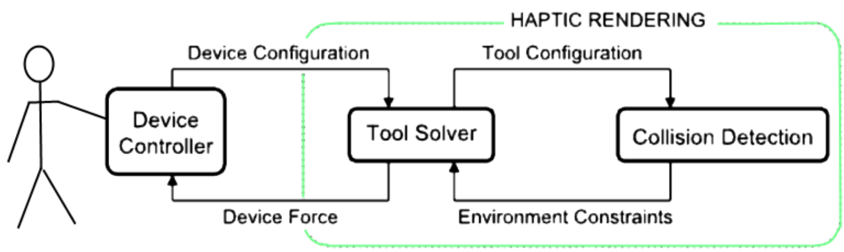

Figure 1.4: Haptic rendering scheme (from [MO08])

1.4 Applications of haptic interfaces

Haptic interfaces are used in various domains [MO08]. A virtual reality system using haptic feedback provides an interesting spatial perception and allows the user to better "understand" the simulation or to immerse him into the virtual world.

Haptic systems are used for example for virtual prototyping, in order to design new three dimensional virtual mechanical objects. Virtual prototyping is common in automobile, aeronautics and even nanotechnologies. This technique can lead to reduced production cycle time but also to decrease costs of conception.

The spatial feedback of haptic devices is also use for scientific visualization (DNA, molecules), and interaction with the materials (DNA unfolding, combining molecules). More generally Haptics can also useful in medical applications, like for example surgery simulation in order to train surgeons with lower costs and risks. In the following figure (1.5) we can see a bi-manual haptic simulation of heart surgery, providing a training for future surgeon without risks for patients.

Figure 1.5: Simulation of a heart surgery (King's College London)

Haptic feedback can also bring assistance to the surgeon by correcting his movements. Haptic device can provide medical help to patients too: rehabilitation make use of various haptic systems such as hand exo-skeleton or ankle haptic stimulator.

Some haptic devices are also used to perform artistic gestures, for example haptic painting or sculpting.

An important field of haptic interfaces called collaborative haptic can be used in various domains. Collaborative tasks are achieved by sharing a virtual world between multiple users using haptic interfaces. This field will be studied later as collaborative haptic interaction have some similarities with bi-manual haptic interaction.

Chapter 2

Related work

2.1 Hardware

Only few haptic interfaces can be used with two hands. We present in this section some studies on bi-manual haptic interfaces. We will try to find unstudied points that could be addressed during the internship. Basically, most systems use two times the same one-hand haptic device as seen on [Ott09] and [EYK10]. Several systems use two simple devices and connect them to separate virtual tools. We chose to detail only interfaces specifically designed for bi-manual interactions.

We classified the systems encountered in two categories: independents interfaces and connected interfaces. The use and problematic of these categories are different, mainly for the workspace management.

2.1.1 Two independents interfaces

Haptic-Workstation

The thesis of [Ott09] describes a two-handed haptic interface called Haptic-Workstation. The system shown in the figure 2.1 consists in a seat with two robotics arms (Cyber-Force) used as arms exo-skeleton.

Figure 2.1: Bi-manual Haptic-workstation system (from [Ott09])

At the end of each arm is connected a haptic interface called CyberGrasp, simulating forces on fingers by tendons connected to fingerprints and motors. Fingers positions are calculated with the CyberGlove data-glove. Haptic-workstation is a symmetrical device with 22 DoF if we count 1 DoF per finger. The system is used in a three dimensional environment exploration with bi-manual interaction with objects in it.

This interface has a large workspace but does not consider workspace collisions. The designed framework contains a physical engine, a haptic thread and an abstraction layer. The conception of our framework is partly based on that work.

Bi-manual Hiro III

As described in [EYK10], HIRO III is a haptic device consisting in a robot arm with a robotic hand connected to the five fingers of the user hand. The system is a mirrored robotic limb (upper arm, lower arm, wrist, hand) of the user limb. This interface can provide three-dimensional directional forces on each finger. The bi-manual haptic device is obtained by putting side to side two HIROs III and connect them to two computers connected with a fast network (TCP Gigabit channel). In the paper the authors point out some workspace issues, for example workspace collisions. Bi-manual Hiro III is a symmetrical device with 42 DoF. They test this system on a bolt and nut simulation with a screwing-turning task.

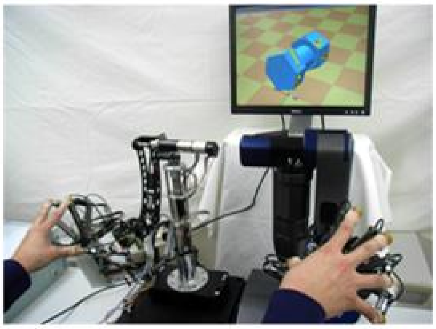

Figure 2.2: Bi-manual Hiro III system (from [EYK10])

Although these two systems were using separate interfaces, the two haptic interfaces were similar. The physical workspace is separated in two symmetrical workspaces separated by a plane.

2.1.2 Two connected interfaces

8-fingers Spidar

The Spidar system presented in [KWH+01] is based on fingertips connected to each user's finger. Those fingertips are connected to three wires each, pulled by three motors place in three different places. This technique allows three-dimensional directional forces on four fingers per hand. This interface provides good quality force feedback in the workspace but is limited by the strings. This interface as been conceived for a two-hands use contrary to the following systems. Spidar is a symmetrical device with 24 Degrees Of Freedom (3 DoF * 8 fingers). The system is used for example to simulate an interactive Rubik's Cube.

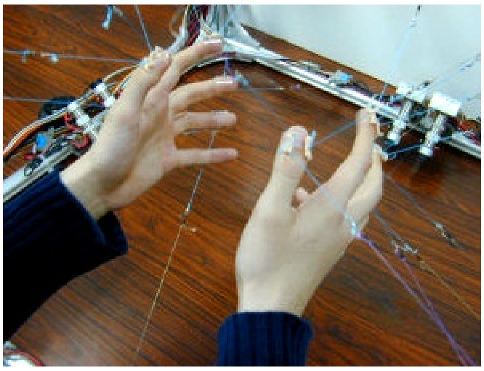

Figure 2.3: Bi-manual Spidar system (from [KWH+01])

Mobile Haptic Interface

The mobile haptic interface by [PKB07] is an innovative system that uses two robotic arms connected to a mobile platform. The system can move around the room while providing bi-manual haptic feedback. This solution permits to extend the small workspace of the robotic arms to the size of a room. The study focus on the conception of the platform, the kinematic and position tracking. The workspaces are separated by a plane in order to avoid collision between the two arms.

Figure 2.4: Bi-manual mobile system (from [PKB07])

These two systems provide interesting solutions but can be limited. The workspace of the Spidar is limited to the cube containing the wires. The bi-manual mobile system solve the workspace problem but bi-manual interaction has not been deeply studied yet.The two haptic systems separate the two hands by a plane.

2.2 Software

As seen in figure 1.4 haptic devices must be connected to a calculator processing the haptic rendering. For bi-manual devices the rendering must calculate a force feedback for the two hands. Algorithms implementing the rendering for bi-manual haptic devices are mainly based on two uni-manual haptic rendering algorithms. The Figure 2.5

presents the hardware-software connection through a process called virtual coupling. The tool solver bloc represents the interaction technique between the haptic interface and the virtual world. Finally the collision blocs handle the physical properties of the scene concerning object collisions and intersections.

2.2.1 Virtual coupling

One important topic of haptic software is virtual coupling. It consists in simulating one virtual spring and damper between the haptic interface position and the virtual cursor in the virtual world. Virtual coupling is used to stabilize the simulation as seen in [MO08] but can be also used to transform a change in position into a force and the opposite.

The Figure 2.5 shows a virtual coupling for one device but can be extended for two.

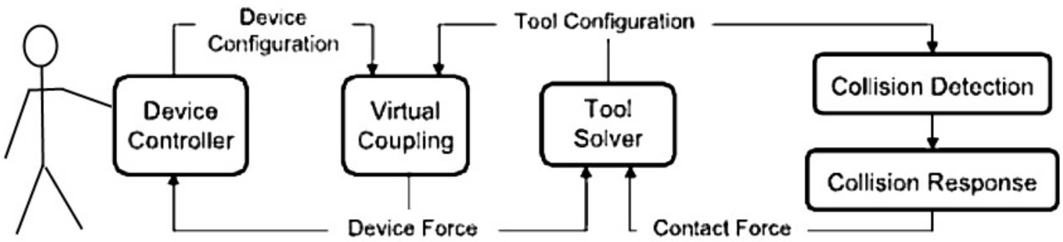

Figure 2.5: Virtual coupling between a haptic interface and a virtual world (from [MO08])

However, virtual coupling can be more complex as we have two interactions in the virtual world instead of one. The virtual coupling for collaborative haptic interaction proposed in [MO08] can be used for bi-manual haptic rendering when the two hands are manipulating a single object. A virtual hand can be used as a virtual cursor [Ott09] making the simulation more realistic but the virtual coupling more complex to handle.

2.2.2 Physical simulation

When more than one object is present in the virtual world, the rendering algorithm have also to simulate collisions. As seen in figure 2.5, the collision simulation is done in two steps: collision detection and collision response.

The collision detection phase can be realized by several methods.

- Voxel based [MPT99]

  The virtual world is sub-divided into small volumetric pixels

- Level of detail [OL05]

  The collision detection is done with several levels of detail in order to accelerate computation

- Constraint based [ORC07]

  Physics constraints are applied to objects in the virtual world in order to achieve better quality results.

The collision response is mainly realized with penalties based methods. The force feedback is proportional to the distance of penetration between objects. In [EYK10] this method is used to calculate the force feedback generated by a virtual hand pressed against a table. The forces are applied to the virtual hand, then to the haptic interface.

The methods described above are expendable to bi-manual interaction in virtual worlds as they can manage more than one collision simultaneously.

Several physics engines realize the collision detection and response using algorithm seen above. They also simulate physical interactions such as gravity, friction, soft materials, fluids, etc. The three principal physics engines are Havok, Bullet and PhysX all available for academic uses.

2.2.3 Interaction techniques dedicated to bi-manual manipulations

An interaction technique is a combination of hardware and software elements that provides a way for users to accomplish a single task. Few bi-manual haptic interaction techniques have been studied in the field of haptic interaction. We have seen that many interfaces have a limited workspace and software solution exists to overcome this issue. A solution used by [Ott09] with the haptic workstation consists in holding out the two arms in a direction, thus moving a virtual body through the scene. We did not find many other haptic interaction techniques with two hands. However we can find some inspiration on related fields.

[KW05] proposes to use the two hands without any active device. They only track the hands and immerse the user in a virtual world. One hand defines a frame of reference for the other hand that act on the surface of the first hand. When the acting hand presses a button on the virtual world it presses the “reference hand” giving some tactile feedback to the user.

The tactile interaction field is another field of inspiration. Several studies such as [Wil04] proposes two-hands interaction techniques and metaphors for selection, positioning or scaling.

Studies in human perception can also be taken into account in order to make a better conception of interactions. The work of Guiard in [Gui87] proposes clear classification of human bi-manual tasks and specificities of each hand. He makes the distinction between the preferred hand and the non-preferred hand. The preferred hand makes movements with less amplitude but more precise. For example when handwriting the preferred hand makes the precise and small movements, the other hand moves the sheet of paper in sweeping gestures. The non-preferred hand starts the bi-manual moves before the preferred hand.

2.3 Conclusion

We have seen that several bi-manual haptic interfaces have been developed, from the more simple to the more complex systems. We compare them with several criteria:

- Symmetry

  Is the device composed of two similar haptic interfaces?

- Workspace size

  Is the reachable physical workspace limited?

- Workspace collision

  Can the haptic device collide between them or with the user.

- Bi-manual specific

  Is the devices composing the bi-manual interface specifically conceived for bi-manual haptic interaction

Table 2.1 assesses the different specificities.

System	Haptic Workstation [Ott09]	Hiro III bi-manual[EYK10]	8-finger Spidar [KWH$^+$01]	Mobile haptic interface [PKB07]
Category	Two independent interfaces		Two connected interfaces	
Symmetric	**yes**	**yes**	**yes**	**yes**
Workspace size	**limited**	**limited**	**very limited**	huge
Workspace collisions	**yes**	**yes**	no	**yes**
Bi-manual specific	**no**	**no**	yes	**no**

Table 2.1: Recapitulation of bi-manual haptic interfaces specificities

First we notice that all studied interfaces are symmetrical. We can see that the workspace management is very important. Only the mobile haptic interface gets a huge workspace. Among the others, only the haptic workstation proposes a software interaction technique to virtually extend the workspace. One important point is the collision between the two physical workspaces. Only the 8-finger Spidar do not encounter this problem due to mechanical properties of the device and no interaction techniques. It is also the only bi-manual specific architecture.

The studies on bi-manual haptic interaction are very limited and the challenge of workspaces (both real and virtual) management is far to be mastered. There is also not many innovative haptic techniques associated with bi-manual tasks.

In order to develop these techniques we need a bi-manual haptic environment. In the following chapter we will present our bi-manual haptic framework. This environment will use several kinds of haptic interfaces and possibly different kinds at the same time. It will also simulate a virtual world with collision detection and reaction.

Chapter 3

Conception: Hardware to Software

3.1 Description

Based on the bibliography, we need to develop new interaction techniques. In order to develop these techniques in a test environment, we decided to create a bi-manual haptic framework. The framework for bi-manual haptic interaction have to be a tool that could:

- Be compatible with several haptic interfaces
- Provide simple control for those haptic interfaces
- Handle asymmetric bi-manual haptic systems (two different interfaces)
- Simulate a 3D virtual world
- Compute physical simulation
- Help to develop innovative interaction techniques

Therefore we conceived and realized several software modules, combined in a "Bi-manual Haptic Framework", allowing us to test new interactions.

The purpose of this framework is to provide a simplified use of bi-manual haptic manipulation and a compatibility with a panel of interfaces, easily extensible. Then it has to simulate a virtual world, with collision management.

3.2 Hardware configuration

The first demand of this study was to work with several models of haptic interfaces and eventually two different interfaces simultaneously.

The haptic interfaces at our disposal are the Phantom-Omni (Sensable), the Falcon (Novint) and the Virtuose6D (Haption). These interfaces are different on many aspects, their specificity are described in the table 3.1.

The Control type column contains the two main modes of control for a haptic interface. The control mode defines what information the haptic interface send to the haptic simulation and with what unit the calculator controls the haptic interface. The impedance control consists in getting the position/orientation as an output and send forces to apply on the device.The admittance control is the strict opposite, we get as output the forces applied on the haptic interface and we control the position of this interface.

	Max. force	Workspace	DoF Input	DoF Output	Control
 Falcon	8.8N	10*10*10cm	3	3	impedance
 Phantom	3.3 N	16*12*7cm	6	3	impedance
 Virtuose6D	35N	70*45*45cm	6	6	admittance

Table 3.1: Characteristics of the different haptic devices used during the internship

In the table we observe a diversity of capacities we must take into account in order to propose a functional and convincing bi-manual interaction.

However, a generic haptic interface model can be obtained by setting parameters defining any kind of haptic device. We can also think about a solution to unify the control and feedback of these haptic interfaces in order to make some abstraction of the hardware differences. Beyond the hardware-related issues, the bi-manual haptic framework must permit an interaction with a virtual world. For that purpose, we need to have the same control type for the interface. We chose to use the admittance mode by changing the Phantom and Falcon control type.

In order to simulate the virtual world, in addition to the visual representation of the scene, we also need to calculate the haptic feedback for each interface. For that purpose, collision detection must be coupled with a calculation of the haptic feedback and consequently the change in position of the objects concerned.

We will discuss these issues in two parts. First we will describe the parameters of our haptic interface model and the second part will deal with the parameters to unify in order to make an abstraction of hardware specificities

3.2.1 Haptic interface model

We define a generic haptic interface model that constitutes the base of our Bi-manual Haptic Framework. The figure 3.1 shows the model used in the framework. We classified the data in three categories:

State data

All the information defining the current state of the device.

Parameters

Settings values, characteristic to each kind of interface device.

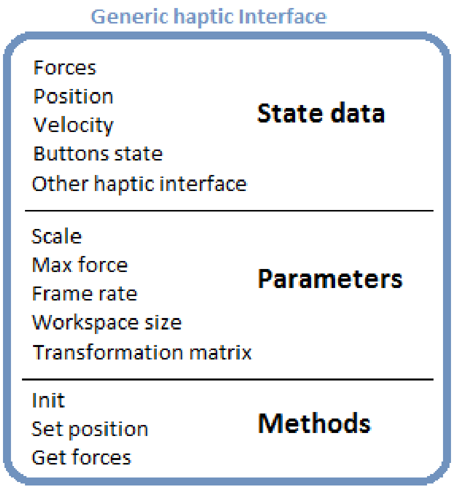

Figure 3.1: Generic haptic interface

By choosing the admittance control mode we must have the position to set and the forces applied on the interface. The speed of the interface is also necessary for several calculations. Most interfaces have at least one button, we store the state of one button. One other parameter is the size of the available workspace that allows to know if the haptic interface has reach the limits of the reachable space. As we work with two interfaces, we also store the other interface used. This permit to realize calculation taking into account the two haptic interfaces.

In the parameters, one important data is the scaling factor. We multiply the dimensions we want to scale by this scaling factor. There can be several dimensions concerned by a scale modification. Thus we store a scale force and a scale position. Another parameter in the transformation matrix that defines the offsets in position and rotation to apply to the real position. The initial value of the transformation matrix is set to the position of the haptic interface in global coordinates. The transformation matrix allows to have more control laws than just position control without scaling.

For the moment, only these specifications are used, however we planned to quickly include other information mandatory for the rest of the internship. For example, knowing the 3D structure of the haptic interface will allow us to calculate the positioning and the space occupation of the haptic interface.

3.2.2 Hardware configuration

When using two different haptic interfaces the feedback provided can be asymmetric, for example the maximum force supported by the device can be different thus making a heavy object perceived lighter on the weaker interface. The differences can cause a lack of immersion in the simulation. We propose a non-exhaustive list of asymmetries:

- Maximum force feedback reachable by the device
- Range of forces the interface can simulate / the haptic resolution
- Latency (between command in forces and feedback)

- Frame rate (e.g. 1000Hz) resulting in more or less stiff rendering
- Stiffness of the device (play in the mechanical parts)
- Size of the workspace
- Input Degrees of Freedom
- Output Degrees of Freedom

The framework address several problems caused by these asymmetries. Some of the above parameters can be obtained automatically (Frame rate, max force), some have to be entered manually. Based on this parameters we can homogenize the output.

We propose some solutions to the above issues. The force feedback is scaled depending on the device that has the lowest maximum force thus limiting the force range. Concerning the frame-rate, the solution of limiting the fastest update time to the lowest one will work but the precision of the feedback will highly decrease. To our knowledge, effects of asymmetric haptic resolution or asymmetric latency have not been studied yet, this could be a trail for future work on that subject. The degrees of freedom may be limited to the more limited haptic interface but it is highly dependent on the type of interaction.

3.3 Software configuration

3.3.1 Global view

The figure number 3.2 shows the organization of the different blocks.

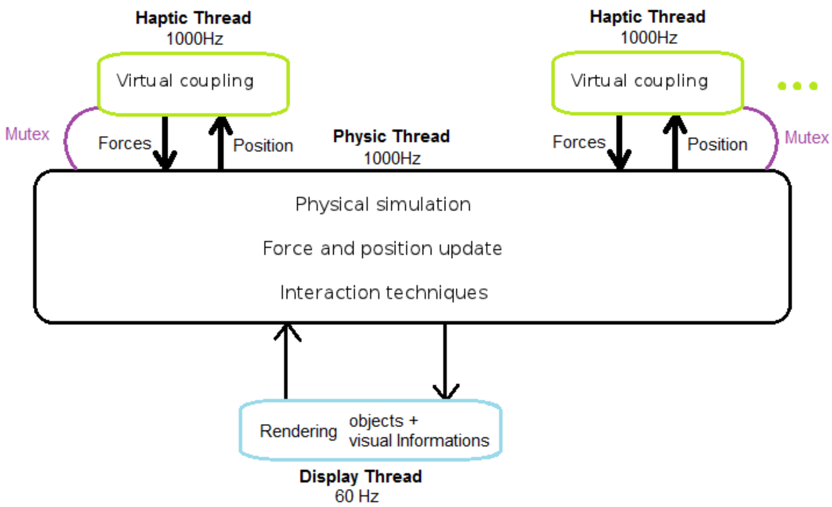

Figure 3.2: Organization of the haptic interface framework

A haptic thread is created per interface connected. It contains the initialized parameters of the generic haptic interface. The physics thread is running at the same frequency as the haptic threads. As the haptic threads and the physic thread share some variables (position and force) we used a mutual exclusion algorithm based on P and V semaphores. The application is shown in the following pseudo-code (Table 3.2). When a P(X) operation is done by one thread, every other thread calling P(X) will wait until the V(X) operation is called. Thus only one thread can access a P(X)-V(X) zone at the same time. In our case, H1 and H2 data are updated separately in haptic threads and in physic thread.

Haptic thread 1	**Haptic thread 2**	**Physic thread**
P (Haptic 1)	P (Haptic 2)	P (Haptic 1) P (Haptic 2)
Critical section update H1 data	*Critical section update H2 data*	*Use of H1 and H2 variables*
V (Haptic 1)	V (Haptic 2)	V (Haptic 1) V (Haptic 2)

Table 3.2: Pseudo code of P and V semaphores ensuring no concurrent access of data

This code is implemented in C/C++ and uses haptic libraries from Novint, Sensable and soon Haption. The simulation runs on a Quad-core that handles parallelism, 64 bits, Intel Xeon CPU with 6 Go of Ram. The configuration allows all the thread to run to the desired speed (1000Hz, 1000Hz and 60Hz, see figure3.2). We generate a scene composed of one plane, two cubes used as manipulating tools, a landscape of static cubes to situate the view and small cube generator.

All objects collide each other and the haptic feedback is calculated on the two cube-tools. The user can feel each element of the scene and interact with the small cubes.

3.3.2 Physical simulation

Physics engine

Haptic rendering in a virtual world implies a collision detection and a collision response. We used an existing physic engine of Nvidia called PhysX to simulate these steps.

PhysX allows fast collision detection and can quickly be integrated with a 3D rendering [Pe]. In addition to collision this physic engine simulate gravity, fluid, soft bodies and force fields. With all these tools it is simple to obtain an interactive and realistic virtual world.

We run a physic thread at 1000Hz realizing the following steps:

1. Simulate one discrete step of the virtual world (1 millisecond)

2. Update the two haptic interfaces position

3. Update the position of the tools connected to the interfaces (for example cubes)

The admittance control of the haptic interface is necessary because manipulating a tool by directly changing its position (discrete steps) will cause some incoherence such as a solid object going through another one. Solving these incoherences result in huge forces in the objects ejecting them in a non-realistic manner.

Virtual coupling

In order to work with all kinds of haptic interfaces but with admittance control only, we need to provide an abstraction that transform impedance interfaces in admittance interfaces.

The method to realize this transformation is called virtual coupling. As seen in the figure 3.3 it consists in simulating a spring-damper connection between the impedance and the admittance cursors.

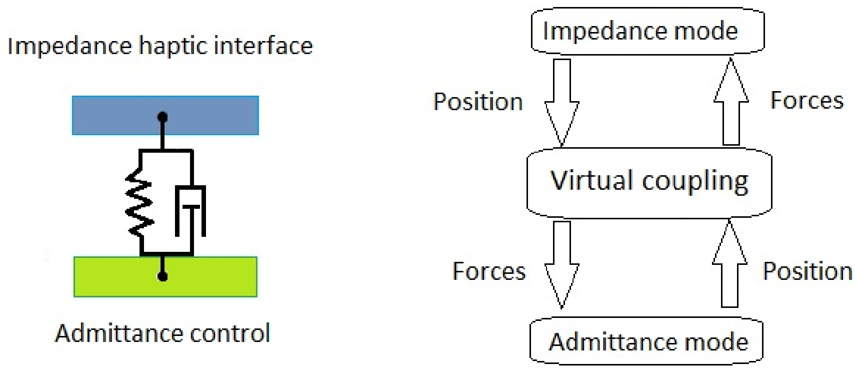

Figure 3.3: Virtual coupling spring and damper

This method allows also to smooth the force feedback. The method is transparently used in the bi-manual haptic framework, a virtual coupling step is realized at every position/force update of impedance haptic interfaces. The virtual coupling is realized independently on each interface and does not need an adaptation

The parametrization of stiffness and damping is no trivial. Too much stiffness and the forces may be too important and may cause vibrations, not enough and the virtual cursor will act like as if it was connected to a yielding elastic and may oscillate when the user stops brutally its movement. The damping is highly dependent on the stiffness and control the oscillation. Not enough damping and the cursor will oscillate, too much and the cursor will move as it was in a viscous environment. These factors are completely dependent on the haptic device but also the scaling factor. To our knowledge, no studies determines a precise setting for stiffness and damping during scaling changes.

A work-in-process is to change the position scale as a function of distance between cursors. The virtual coupling adaptation is not established yet and the haptic feedback can become unstable due to the scale change.

3.4 Conclusion

We set an environment compatible with three kinds of haptic interfaces. We can easily control all of them and create bi-manual haptic systems composed with two different haptic devices. We access all haptic interfaces the same way in order to be compatible with the physics engine. The physics simulates collision and is integrated in the 3D virtual world.

Using this framework we can now develop new interaction techniques. The objective is to obtain a software solution to use two haptic interfaces simultaneously on different tasks.

Chapter 4

Interaction techniques

4.1 Scenes for bi-manual haptic interaction

Using the framework we can realize virtual scenes in order to test our bi-manual haptic interaction techniques and possibly find new problems to solve. We describe here nine scenes, categorized into three categories depending on the interaction type (categories defined in [Gui87]).

4.1.1 Serial Interaction

Only one hand interact at the same time

One Object: Pick & Place Uni-manual

Scene: a cube in a scene (that could be a plane).
Manipulation: pick the cube with 1 hand to put it on another place. The cube must be picked with the left interface then the right, etc.

Two Objects: Weight Comparison

Scene: two objects with a weight.
Manipulation: The two objects are connected to the respective haptic interface. The user has to move the two objects up and down in order to determine the witch object in the heavier (or just feel the pressure in his hand). The two objects must be weighted one after the other (or simultaneously in the parallel mode).

One Object, One Scene: Control scene & object

Scene: cube in a scene (that could be a plane).
Manipulation: One haptic interface controls the object with 6 DoF and the other controls the world/the scene.

4.1.2 Parallel Interaction

Two hand acting on the same dimension(s) at the same time

One Object: Pick & Place Bi-manual

Scene: a cube in a scene (that could be a plane).
Manipulation: pick the cube with the two hands to put it on another place. The cube is initially in the workspace of the first interface and must be moved to the workspace of the second interface.

Two Objects: Peg in a Hole

Scene: a peg and a cube with a circular hole.
Manipulation: place the peg in the hole. The peg is connected to an interface and the cube to the other.

One Object, One Scene: Control Scene & object

Scene: cube in a scene (that could be a plane). Another scene could be an object placed on the surface of a globe (that is the scene)
Manipulation: the scene can be translated 3DoF and the object too. If the user moves the scene, the interface connected to the object tend to move the same way (stopping it causes the object to move in the opposite direction). The opposite situation (moving the object in the scene) does not move the scene or the interface connected to the scene.

4.1.3 Separate Interaction

Two hand acting on independent dimension(s) at the same time

One Object: Independent axis manipulation

Scene: a cube in a scene (that could be a plane).
Manipulation: a haptic interface controls the Z axis and the other the X and/or Y axis. The forces produced by the interfaces can only be on the axis they control so they can only be caused by the collision between the object and the scene and not by interaction between the two interfaces.

Two Objects: Pencil & Sheet of paper

Scene: a pencil and a sheet of paper on a plane.
Manipulation: the pencil is connected to the preferred hand haptic interface (if possible the phantom) and the other to the sheet of paper that can be moved on the surface of the plane (2 translations 1 rotation). The user can write on the sheet of paper with the pen.

One Object, One Scene: Control Globe & object

Scene: an object placed in a scene (that could be a plane).
Manipulation: the scene can be rotated with 3DoF and the object translated with 3Dof. The rotation of the scene is object centered and so does not affect the object (not forces transmitted to the object)

4.2 Challenges

In order to determine efficient interaction techniques, we need to take into account the main challenges raised by the use of two haptic devices simultaneously. We defined three main topics to address while designing our interaction techniques:

- Virtual and real workspace management

  Real workspace management is not necessary if we work with one interface. When working with two interfaces, physical collisions can occur as seen in figure 4.1.The zone where collisions can occur is the intersection between the two workspace zones.

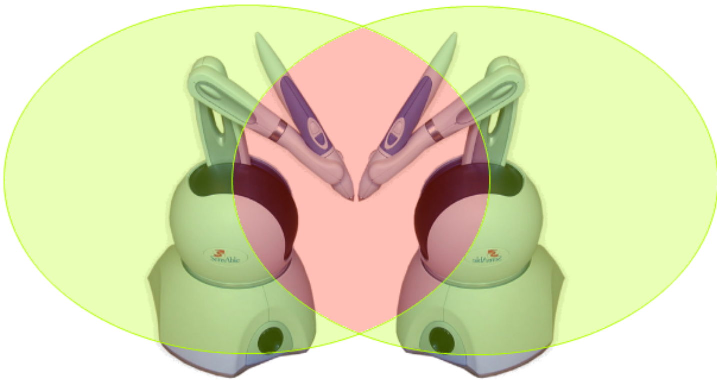

Figure 4.1: Intersection of two haptic interface workspaces

  Virtual workspace depends on the scale between control (C) and display (D) called C/D ratio. Working with no scale between the real interface movements and the virtual scene displacement allows a good precision and immersion for the user. However the workspace provided by small haptic devices like Falcon or Phantom will be too limited for an efficient bi-manual interaction. In addition to scaling, we can use command laws like control in speed to extend the virtual workspace.

- Genericity with different haptic interfaces

  Using two different interfaces can cause asymmetries in the interaction. The interaction techniques should work with different kinds of device.

- Ergonomics of the interaction

  We focused on one interaction: the picking task. As we will work with scaling the user will loose some precision. We need to provide some tools to facilitate the interaction with the virtual world. Grasping a cube with two tools and move it is not trivial with scaling as a small movement makes the cube to fall. We propose a simple solution to avoid this problem based on contact detection.

4.3 Different interaction techniques

Contrary to the mobile platform used in [PKB07], we only use grounded interfaces. The physical workspace provided is limited and when the virtual cursors are controlled in position, the virtual workspace has the same limitations. To address this issue, some interaction techniques have been proposed. The positioning of the virtual pointers can be obtained using different command laws. We focused on the three following techniques:

1. Position (With C/D ratio) :
   Consists in using the position of the haptic interface multiplied by a scale factor to define the position of the virtual cursor

2. Clutching:
   When reaching an uncomfortable posture or the end of the workspace, the connection between the scene and haptic interface can be released (for example by a button) allowing a re-centering of the haptic interface without moving the scene.

3. Speed:
   Calculates the speed of the virtual cursor proportionally to the displacement of the haptic interface around a center zone, just like a joystick.

A interesting haptic interaction is called the Bubble [DLmB+05]. This metaphor takes the advantages of the three previous interaction methods but is only proposed for one-hand haptic interaction. When the haptic interface in within a defined sphere, the virtual cursor is controlled in position. If outside, the cursor is controlled in speed. The bubble is represented on the 3D scene as a semi-transparent sphere indicating the boundaries of the position control zone. A haptic force-feedback is given to the user to make him perceive the bubble surface.

Although this last interaction technique is efficient and intuitive with one haptic interface, using two bubbles on a bi-manual interaction is tricky. Indeed, the user does not understand easily the functioning of two bubbles at the same time. The surface of the bubble is reached faster and with less control as two semi-transparent spheres are displayed on the screen causing the user to pay attention to none of them. The user can also lose the two virtual cursors if moved in opposite directions with speed control because the camera can not follow the two cursors simultaneously. During a precise task like picking an object one more issue appear. If one haptic interface reaches the surface of the bubble before the other one, only one cursor will be controlled in speed thus causing an asymmetric command law. The task become then much harder and the picked object may fall.

4.3.1 A new interaction technique based on Bubble metaphor

We conceived a new metaphor, inspired by the bubble metaphor. First we used a 3D parallelepiped instead of the sphere. The workspace used with position control is in most of the cases more filled with this shape. When reaching the extremities of the "brick" the virtual cursor is also controlled in speed as shown in figure 4.2.

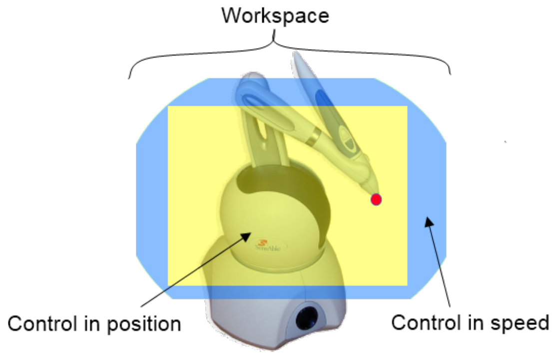

Figure 4.2: Illustration of the “brick” in the workspace

We decided not to give force feedback of the surface to the user in order to simplify the sensorial information as the shape is less simple than a sphere. The visual feedback has been modified too, instead of a transparent bubble, the tool simply blurs when reaching the extremities of the shape. This effect looks like motion blur caused by the speed as seen in figure 4.3.

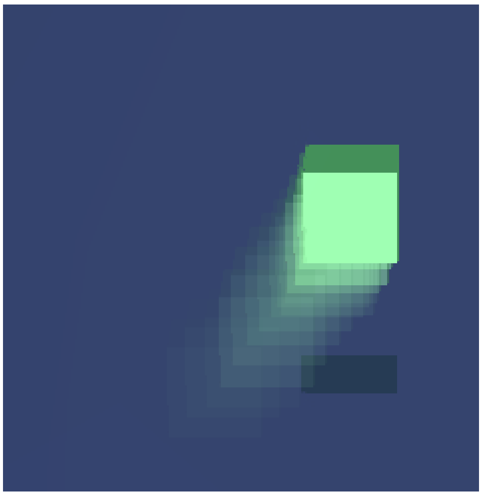

Figure 4.3: Motion blur for signaling speed control metaphor

The brick shape could be modified to a more complex convex shape in order to use the maximum of space in position control.

4.3.2 Camera control

Similarly to clutching, speed control and the bubble metaphor, our technique provides an infinite workspace reachable by the virtual cursors. In order to always have the cursors in sight we had to control the camera.

When the two cursors move aside, the camera simply follows the center between the cursors. When one cursor is moved away from the other cursor, the camera move backward in order to continuously see the two cursors. The following algorithm and scheme defines the 3D camera position.

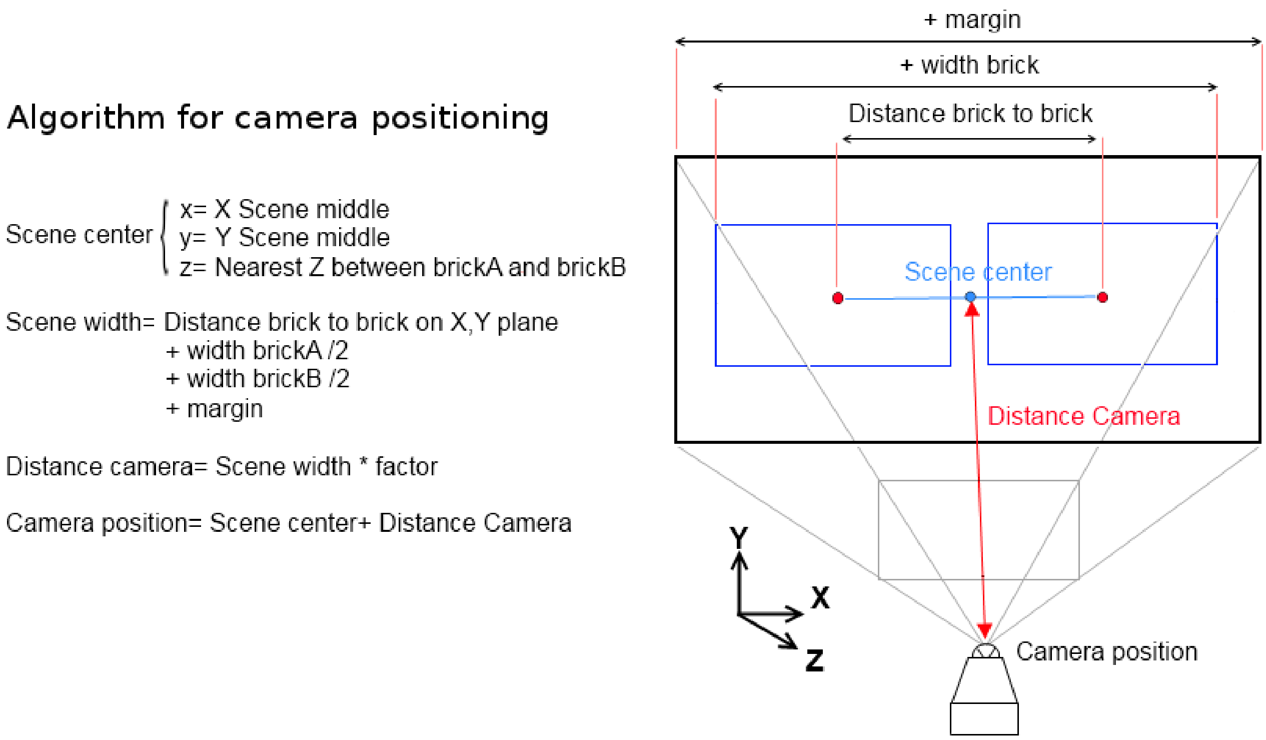

Figure 4.4: Algorithm and scheme of camera positioning

The scaling could also be modified in order to maintain a constant ratio between control and display.

4.3.3 Physical workspace management

The workspace of both interfaces can physically intersect as seen in figure 4.1. This can cause collisions between haptic interfaces witch can be perceived by the user and reduce immersion. Collision between interfaces could also damage the hardware. Modeling the 3D shape of each interface can permit to reduce physical collision by creating, for example, repulsion between the two interfaces (e.g. same pole magnet). We will use a simplified 3D model of the haptic interface coupled with the position and orientation. As seen in the figure 4.5, we can use one bounding box per robotic axis.

Figure 4.5: Five bounding-boxes around the phantom haptic interface

Several levels of detail could be set, from the exact 3D representation to the single box. The simplest way to avoid collision is to separate the two interfaces by a plane positioned between them taking into account an approximation of the workspace.

One other problem could be collisions between one haptic interface and the user hand/arm. When consequent forces are applied on the devices this could cause serious injuries. This situation is more complex to deal with because unless we dispose of a tracking system we can not determine where the user is. One solution could be to use an approximation of all the possible places like a cone pointing on the haptic interface.

4.3.4 Ergonomics of interactions

Ergonomics is crucial in user interfaces in order to avoid discomfort or fatigue. Haptic interactions need to be user-friendly, more precisely easy to understand and effortless to use.

We developed several visual and haptic techniques to improve the user experience. Virtual cursors are currently represented by one blue cube and one green cube. The color differentiation is a simple way not to confuse the two cursors. However, we noticed that some users had troubles to associate the cursors to the right haptic interface when the cursors were crossed and not the haptic interfaces. This situation illustrated in figure 4.6 is possible due to scaling factor, the bubble metaphor, clutching or our interaction technique.

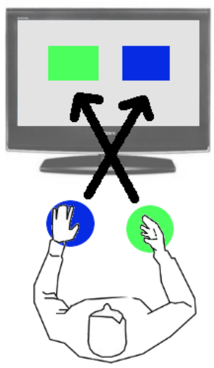

Figure 4.6: Inversion between the user posture and the visual simulation

A future change will be to design two tools of different shapes similar to hands or claws in order to better identify the direction/association. Simple shadows are rendered. Projected at the vertical of every objects, they help the user to locate their depth and improve the positioning accuracy [Hubona 2004].

When the two cursors are in contact or manipulate the same object, dealing with two potentially different scales and control type (position, speed) the user may lost the contact because of the non-trivial asymmetry. We decided to switch to a cooperative mode in these situations. We represent this mode change by connecting the two cursors by a red arc.

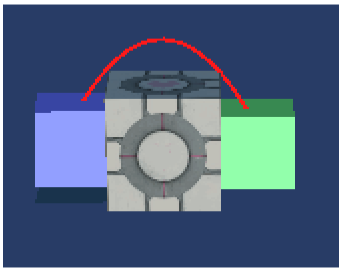

Figure 4.7: Connected cursors when manipulating an object

The cooperative mode consists in imposing the same C/D ratio and the same control type. If one of the cursor is speed controlled , the second one become speed controlled too. The speed of each cursor is set to same value, the mean of speed they would have if they were not connected.

We finally give a force feedback to the user simulating a spring between the two cursors, making the separation of the two devices a little harder. As soon as the contact stop, the force stop. The perceived haptic feedback is close to a magnet attraction on contact.

One optional help for the pick task is to apply a force on the picked object, pulling it with a small force between the two interfaces. This helps to avoid drops but it decrease the physical realism of the scene.

4.4 Conclusion

We drawn up a list of scenes for bi-manual haptic interaction that will be use for future evaluation of our interaction techniques. For the pick and place task, we developed a new interaction technique based on Bubble metaphor and a camera control is used to follow the cursors.

We started the realization of a physical workspace management, based on bounding boxes. We finally studied ergonomics of bi-manual interaction, more precisely on pick and place task. We developed visual and haptic techniques to help the user in the task and provide more comfortable use.

There is still some work to do on interaction techniques, more particularly on interaction with the other scenes described in the list of section 4.1. The physical workspace collision avoidance have to be integrated too. The bi-manual haptic interactions have to be tested, that is why we plan to realize a study with volunteer users.

Chapter 5

Conclusion and perspectives

The purpose of this work is is to study bi-manual haptic interaction in virtual world. We have seen that existing work on that subject could be improved. More precisely we noticed scientific lacks concerning generic usage of multiple haptic devices and bi-manual interaction techniques. We pointed out four challenges to address:

- Compatibility with several haptic interfaces and asymmetric association of bi-manual devices
- Real workspace management in order to avoid physical collisions
- Realistic virtual world simulation with workspace extension by an interaction technique
- Conceive new bi-manual haptic interaction techniques

We realized a framework for bi-manual haptic interaction in virtual world. This framework is compatible with several kind of haptic interfaces and can take asymmetric bi-manual systems. The framework also includes a 3D virtual world simulating physics and collisions.

Several interaction techniques have been developed during this internship:

- New bi-manual haptic interaction technique inspired by the bubble technique: This interaction technique keeps an important part of the physical workspace to control the virtual cursor in position. The visual and haptic feedback are simplified in order to provide an easily understandable interaction.
- Camera control: In order to fit the view to the cursors position on the screen, we control the position camera. We also proposed to set a factor scaling proportional to the distance between interface
- Haptic assistance for picking task: We realized a picking task assistance. A visual information is displayed, informing the user that a connection has been made. We also add a small force to the picked cube that attracts it between the two cursors. This little attraction facilitates the manipulation without disturbing the task.

For the rest of the internship we plan to experiment the usability of the developed techniques. First we will have to set up test scenes that implement interaction tasks. We also need an evaluation process taking into account the efficiency of the interaction and the comfort.

Finally we will realize our experimentation with a panel of volunteers. The purpose of this user study is to evaluate the usefulness of our work and to spot the weaknesses of the bi-manual interactions.

Bibliography

[CMHL10] Gabriel Cirio, Maud Marchal, Sebastien Hillaire, and Anatole Lecuyer. Six degrees-of-freedom haptic interaction with fluids. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints), 2010.

[DLmB+05] Lionel Dominjon, Anatole Lécuyer, Jean marie Burkhardt, Guillermo Andrade-barroso, and Simon Richir. The "bubble" technique: Interacting with large virtual environments using haptic devices with limited workspace. In *World Haptics Conference (joint Eurohaptics Conference and Haptics Symposium*, pages 639–640, 2005.

[EYK10] Takahiro Endo, Takashi Yoshikawa, and Haruhisa Kawasaki. Collision avoidance control for a multi-fingered bimanual haptic interface. In *Proceedings of the 2010 international conference on Haptics - generating and perceiving tangible sensations: Part II*, EuroHaptics'10, pages 251–256, Berlin, Heidelberg, 2010. Springer-Verlag.

[Gui87] Yves Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19:486–517, 1987.

[KW05] L. Kohli and M. Whitton. The haptic hand: providing user interface feedback with the non-dominant hand in virtual environments. In *Proceedings of Graphics Interface 2005*, pages 1–8. Canadian Human-Computer Communications Society, 2005.

[KWH+01] Y. Kohno, S. Walairacht, S. Hasegawa, Y. Koike, and M. Sato. Evaluation of two-handed multi-finger haptic device spidar-8. In *Proceedings of the 11th International Conference on Artificial Reality and Tele-existance (ICAT 2001), Tokyo, Japan*, 2001.

[MO08] Lin C. Ming and Miguel A. Otaduy. *Haptic Rendering: Foundations, Algorithms and Applications.* A. K. Peters, Ltd., Natick, MA, USA, 2008.

[MPT99] William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 401–408, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[OL05] Miguel A. Otaduy and Ming C. Lin. Sensation preserving simplification for haptic rendering. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[ORC07] Michael Ortega, Stephane Redon, and Sabine Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. *IEEE Transactions on Visualization and Computer Graphics*, 13:458–469, 2007.

[Ott09] Renaud Ott. *Two-handed haptic feedback in generic virtual environments.* PhD thesis, Ecole polytechnique fédérale de Lausanne, Lausanne, 2009.

[Pe] Nvidia Corporation PhysX engine. http://www.nvidia.com/object/physx_new.html.

[PF06] Alain Berthoz Jean-Louis Vercher Philippe Fuchs, Guillaume Moreau. *Le traite de la realite virtuelle.* Presses de l'Ecole des Mines, 2006.

[PKB07] A. Peer, Y. Komoguchi, and M. Buss. Towards a mobile haptic interface for bimanual manipulations, 292007-nov.2 2007.

[Wil04] Andrew D. Wilson. Touchlight: An imaging touch screen and display for gesture-based interaction. In *In ICMI 04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76. ACM Press, 2004.